

PathCommonPrefix

The destination string buffer must be long enough to hold the return file path

Sean Barnum, Digital, Inc. [vita¹]

Copyright © 2007 Digital, Inc.

2007-04-02

Part "Original Digital Coding Rule in XML"

Mime-type: text/xml, size: 4343 bytes

| Attack Category | <ul style="list-style-type: none">• Malicious Input• Path spoofing or confusion problem | | | | | | | | | |
|---|--|-------------------|------------------------|----------------------|-------------------|---|---|------------------|-------------------|--------------------|
| Vulnerability Category | <ul style="list-style-type: none">• Buffer Overflow• Unconditional | | | | | | | | | |
| Software Context | <ul style="list-style-type: none">• File Path Management | | | | | | | | | |
| Location | <ul style="list-style-type: none">• shlwapi.h | | | | | | | | | |
| Description | <p>The destination string buffer for the PathCommonPrefix() function must be long enough to hold the return file path.</p> <p>The PathCommonPrefix() routine takes two path strings and checks for a common prefix (e.g., "C:\win") then writes that into a buffer. The resulting string could potentially be as long as the shortest of the two strings. The documentation indicates that the output buffer size should be MAX_PATH characters in length.</p> | | | | | | | | | |
| APIs | <table border="1"><thead><tr><th>Function Name</th><th>Comments</th></tr></thead><tbody><tr><td>PathCommonPrefix</td><td>Src: 0, 1.</td></tr><tr><td>PathCommonPrefixA</td><td>Src: 0, 1. ASCII</td></tr><tr><td>PathCommonPrefixW</td><td>Src: 0, 1. Unicode</td></tr></tbody></table> | | Function Name | Comments | PathCommonPrefix | Src: 0, 1. | PathCommonPrefixA | Src: 0, 1. ASCII | PathCommonPrefixW | Src: 0, 1. Unicode |
| Function Name | Comments | | | | | | | | | |
| PathCommonPrefix | Src: 0, 1. | | | | | | | | | |
| PathCommonPrefixA | Src: 0, 1. ASCII | | | | | | | | | |
| PathCommonPrefixW | Src: 0, 1. Unicode | | | | | | | | | |
| Method of Attack | <p>The attacker can overflow the destination buffer if it is not long enough to hold either of the input source strings.</p> | | | | | | | | | |
| Exception Criteria | | | | | | | | | | |
| Solutions | <table border="1"><thead><tr><th>Solution Applicability</th><th>Solution Description</th><th>Solution Efficacy</th></tr></thead><tbody><tr><td>Whenever PathCommonPrefix parameter, is used.</td><td>The third parameter, pszPath, should be at least MAX_PATH</td><td>Effective.</td></tr></tbody></table> | | Solution Applicability | Solution Description | Solution Efficacy | Whenever PathCommonPrefix parameter, is used. | The third parameter, pszPath, should be at least MAX_PATH | Effective. | | |
| Solution Applicability | Solution Description | Solution Efficacy | | | | | | | | |
| Whenever PathCommonPrefix parameter, is used. | The third parameter, pszPath, should be at least MAX_PATH | Effective. | | | | | | | | |

1. <http://buildsecurityin.us-cert.gov/bsi-rules/35-BSI.html> (Barnum, Sean)

| | |
|-----------------------------------|--|
| | characters in length. |
| Signature Details | <pre>int PathCommonPrefix(LPCTSTR pszFile1, LPCTSTR pszFile2, LPTSTR pszPath);</pre> |
| Examples of Incorrect Code | <pre>// String path name 1. TCHAR buffer_1[] = TEXT("C:\\win \\desktop\\temp.txt"); LPTSTR lpStr1; lpStr1 = buffer_1; // String path name 2. TCHAR buffer_2[] = TEXT("c:\\win \\tray\\sample.txt"); LPTSTR lpStr2; lpStr2 = buffer_2; // String path out buffer. TCHAR buffer_3[] = TEXT("abc"); //Note: buffer is too small LPTSTR lpStr3; lpStr3 = buffer_3; // Variable to get the return. // from "PathCommonPrefix" int retval; retval = PathCommonPrefix(lpStr1,lpStr2,lpStr3);</pre> |
| Examples of Corrected Code | <pre>// String path name 1. TCHAR buffer_1[] = TEXT("C:\\win \\desktop\\temp.txt"); LPTSTR lpStr1; lpStr1 = buffer_1; // String path name 2. TCHAR buffer_2[] = TEXT("c:\\win \\tray\\sample.txt"); LPTSTR lpStr2; lpStr2 = buffer_2; // String path out buffer. TCHAR buffer_3[MAX_PATH]; LPTSTR lpStr3; lpStr3 = buffer_3; // Variable to get the return. // from "PathCommonPrefix" int retval; retval = PathCommonPrefix(lpStr1,lpStr2,lpStr3);</pre> |

| | | |
|-----------------------------|---|--|
| Source Reference | <ul style="list-style-type: none"> • http://msdn.microsoft.com/library/default.asp?url=/library/en-us/shellcc/platform/shell/reference/shlwapi/path/pathcommonprefix.asp² | |
| Recommended Resource | | |
| Discriminant Set | Operating System | <ul style="list-style-type: none"> • Windows |
| | Languages | <ul style="list-style-type: none"> • C • C++ |

Cigital, Inc. Copyright

Copyright © Cigital, Inc. 2005-2007. Cigital retains copyrights to this material.

Permission to reproduce this document and to prepare derivative works from this document for internal use is granted, provided the copyright and “No Warranty” statements are included with all reproductions and derivative works.

For information regarding external or commercial use of copyrighted materials owned by Cigital, including information about “Fair Use,” contact Cigital at copyright@digital.com¹.

The Build Security In (BSI) portal is sponsored by the U.S. Department of Homeland Security (DHS), National Cyber Security Division. The Software Engineering Institute (SEI) develops and operates BSI. DHS funding supports the publishing of all site content.

1. <mailto:copyright@digital.com>